SIRPY · Problem 6 (Troesch's problem) — A Famous Benchmark, Zero Tuning, and an Honest Refusal

updated

Luminary Quantum Institute, LLC (dba Mitra Institute of Education)

An ongoing series. We are building a public library of problems SIRPY has solved — each entry is one problem, the exact input we handed the solver, the exact output it returned, and an honest read of what happened. No slides. No hand-waving. Numbers you can check. This entry includes a run that **failed** — and we publish it, because how a solver behaves when it cannot solve a problem is part of what you are buying.

The benchmark

$$y'' = \mu \sinh(\mu y), \qquad y(0) = 0, \qquad y(1) = 1 \tag{1}$$

This is **Troesch's problem**, a classical test case originating in plasma physics — the confinement of a plasma column by radiation pressure — and a standard benchmark for boundary-value solvers precisely because it punishes them.

The difficulty is the sinh of a sinh-like growth: as μ increases, the solution hugs zero across almost the entire interval and then climbs to 1 inside a razor-thin boundary layer at the right end. At $\mu = 15$, the initial slope is on the order of 10^{-6} — the solution starts almost perfectly flat, and everything happens in the last few percent of the interval. Small errors in the initial slope are amplified exponentially across the march. Shooting methods overflow; grid methods need the mesh crowded into a layer they do not know the location of in advance.

We ran it at $\mu = 10$ and $\mu = 15$, and we begin with the simplest possible experiment: **no tuning at all.**

Round 1 — $\mu = 10$, nothing specified

The exact call, with `iterations` and `partition` left out entirely:

```
r = solve_bvp(order=2, f_str="10*sinh(10*y)", x0=0, x1=1,
              left_bc={0: 0, 1: "gamma"}, right_bc={0: 1}
              )
```

What came back:

```
INFO: solved by SHOOTING over a segmented analytic march - each trial
      covers [0, 1] with exact series pieces sized by the radius of
      convergence, and the missing gamma = y^(1)(0) is read off by a
      secant on the terminal mismatch
      (gamma = 0.000358337784721, |y(1) - 1| = 2.15e-11).

INFO: verification by substitution - VERIFIED: satisfies the equation
      to 4.79e-06 on [0, 1] (the equation's scale reaches 1.09e+05;
      max RELATIVE residual, taken pointwise, is 2.48e-10), and the
      boundary data to 2.15e-11.

INFO: exact first-integral test - the solution's singularity is at
      x = 1.00135, OUTSIDE [0, 1] (margin 0.001348).

INFO: run summary - iterations=20 (sirpy default), shooting over a
      segmented analytic march (24 segment(s) in the final trial;
      no partition - n_subintervals is None by design).
```

Table 1: $\mu = 10$, defaults only

Quantity	Value
Recovered $y'(0)$	0.000358337784721
Boundary residual $\lvert y(1) - 1\rvert$	2.15×10^{-11}
Max **relative** residual	2.48×10^{-10}
Absolute residual (equation scale 1.09×10^{5})	4.79×10^{-6}
Segments in the final march	24 (sized adaptively; no fixed grid)
Singularity located	$x = 1.00135$, outside $[0, 1]$

Zero tuning. The solver chose its own resolution, sized its own segments by the local radius of convergence, recovered the missing slope, and delivered a relative residual of 2.5×10^{-10}.

Reading the residuals honestly

Two residuals appear above, and they differ by four orders of magnitude. This deserves a plain explanation rather than a quiet omission.

The **absolute** residual is 4.79×10^{-6} — and on most problems in this library that would be a mediocre number. But $\sinh(10y)$ is enormous near the boundary layer: the equation's own right-hand side reaches 1.09×10^{5}. Measured against the size of the terms actually being balanced, the **pointwise relative residual** — the honest measure of how well the equation is satisfied — is 2.48×10^{-10}.

SIRPY reports all three numbers together: the absolute defect, the scale it should be judged against, and the relative result. A solver that quoted only the flattering number would be marketing; one that quoted only the absolute number would be underselling itself. The point is that you are shown enough to judge.

The singularity, again

Readers of the earlier entries will recognise this line: the solution's singularity sits at $x = 1.00135$ — **just** 0.00135 **past the right endpoint.** That is not a curiosity; it is the *explanation* of the problem's difficulty. The infamous boundary layer of Troesch's problem is the shadow cast on $[0, 1]$ by a pole a hair outside it. SIRPY locates that pole, confirms the interval is genuinely clear of it, and notes that the steepness near the right end is exactly what a nearby singularity produces.

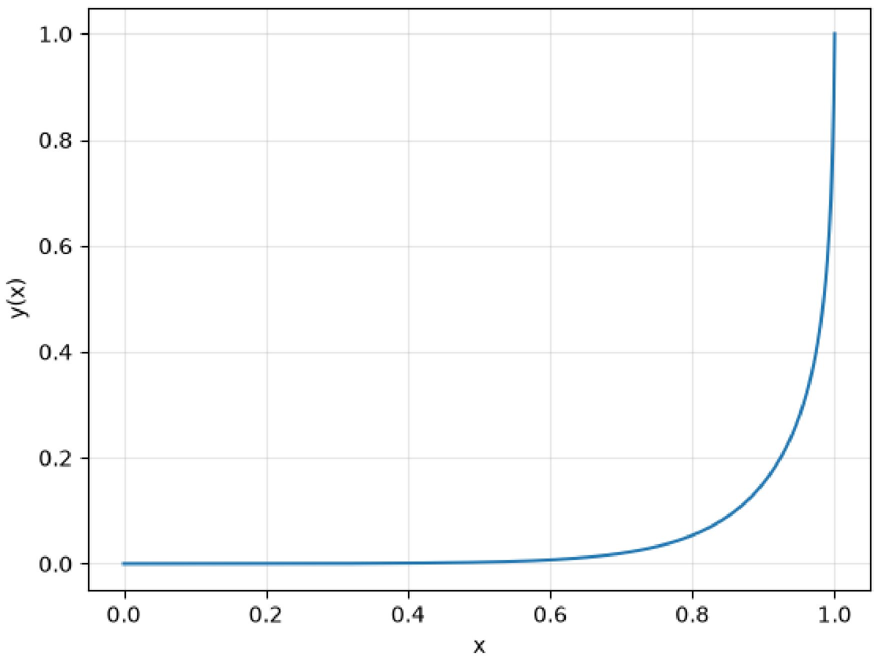
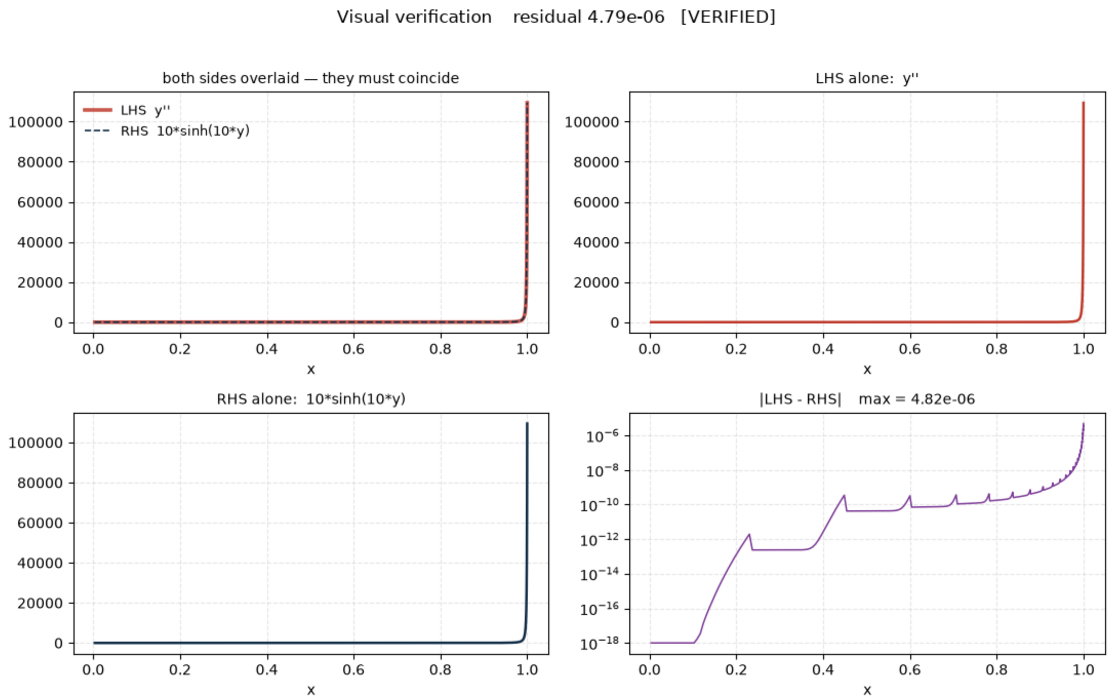


Figure 1: Solution at $\mu = 10$: flat across most of the interval, climbing through the boundary layer to $y(1) = 1$.



Visual verification   residual 4.79e-06   [VERIFIED]
both sides overlaid — they must coincide
LHS y''
RHS 10*sinh(10*y)
LHS alone: y''
RHS alone: 10*sinh(10*y)
|LHS - RHS| max = 4.82e-06
x

Round 2 — $\mu = 15$, nothing specified: the honest failure

Same experiment, harder problem:

```
r = solve_bvp(order=2, f_str="15*sinh(15*y)", x0=0, x1=1,
              left_bc={0: 0, 1: "gamma"}, right_bc={0: 1}
              )
```

What came back — quoted in full, because this is the part of the entry we most want you to read:

```
WARNING: the gamma iteration DIVERGED (iterate values overflowed the
         equation's functions). The returned quantities are the last
         finite iterate and are NOT a solution. Try a different gamma0,
         a shorter interval, or partition=n.

WARNING: partition='auto' found no n <= 64 with good measured matching;
         returning the best attempt (n = 1) - inspect bc_mismatch /
         continuity_mismatch before trusting it.

INFO: verification by substitution - verification failed: this result
      carries no series coefficients to verify.
```

At $\mu = 15$ with default settings, SIRPY **did not solve the problem** — and, more importantly, it did not pretend to. It said, in capital letters, that the returned quantities are **NOT a solution**. Its fallback search reported that nothing it tried met its matching standard. And its verification stage declined to certify a result it could not check.

This is the behaviour we consider the most important feature in the entire library. Every solver fails on some problem; the dangerous ones fail *silently*, handing back a confident table of numbers with nothing wrong visible on a plot. The cost of a numerical tool is rarely the problems it cannot solve — it is the wrong answers it delivers with a straight face. Here, the failure is loud, specific, and refuses certification.

> We are refining this diagnostic so that, instead of listing generic remedies, it recommends the specific setting that resolves this case. The detection and the refusal to certify — the parts that protect you — are shown above, as currently shipped.

Round 3 — $\mu = 15$, one parameter

The fix is a single argument: raise the series resolution.

```
r = solve_bvp(order=2, f_str="15*sinh(15*y)", x0=0, x1=1,
              left_bc={0: 0, 1: "gamma"}, right_bc={0: 1},
              iterations=40
              )
```

```
INFO: solved by SHOOTING over a segmented analytic march
      (gamma = 2.44451302779e-06, |y(1) - 1| = 8.25e-09).

INFO: verification by substitution - VERIFIED: satisfies the equation
      to 3.98e-03 on [0, 1] (the equation's scale reaches 2.39e+07;
      max RELATIVE residual, taken pointwise, is 2.42e-10), and the
      boundary data to 8.25e-09.

INFO: run summary - iterations=40, shooting over a segmented analytic
      march (14 segment(s) in the final trial).
```

Table 2: $\mu = 15$, `iterations=40`

Quantity	Value
Recovered $y'(0)$	$2.44451302779 \times 10^{-6}$
Boundary residual $\|y(1) - 1\|$	8.25×10^{-9}
Max **relative** residual	2.42×10^{-10}
Absolute residual (equation scale 2.39×10^{7})	3.98×10^{-3}
Segments in the final march	14

Note the initial slope: 2.4×10^{-6}. The solution leaves the origin almost perfectly flat — that is how extreme the boundary layer is at $\mu = 15$ — and the secant on the terminal mismatch still pinned it down. The relative residual, 2.42×10^{-10}, is essentially identical to the $\mu = 10$ result, even though the equation's scale has grown by two orders of magnitude to 2.39×10^{7}.

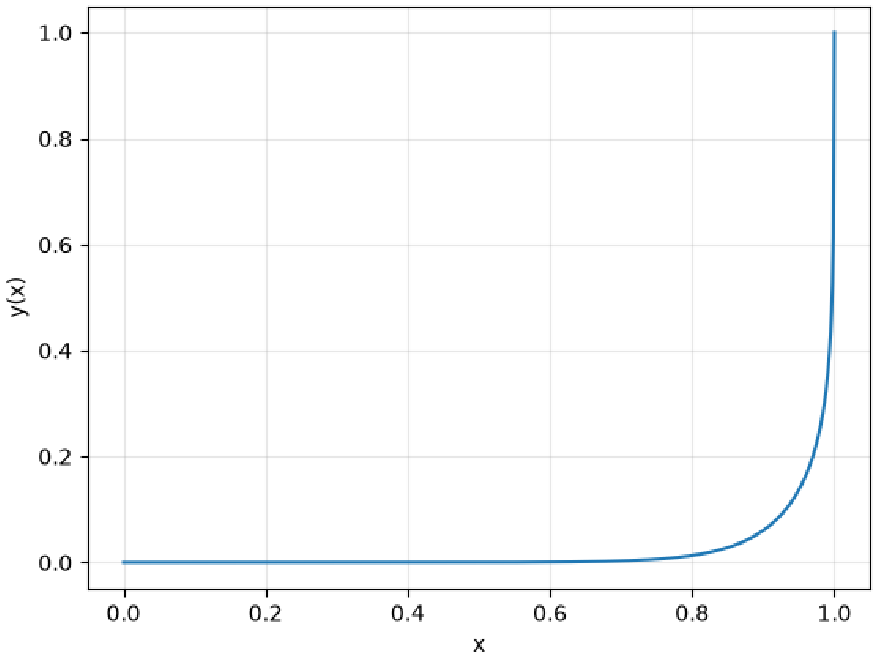
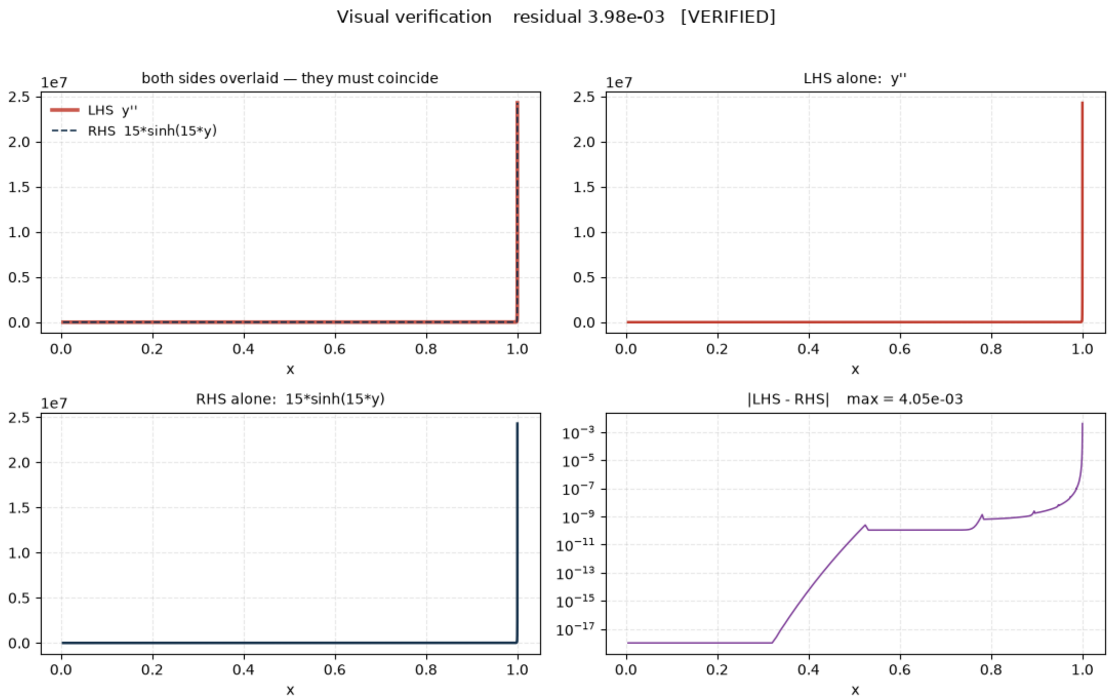


Figure 2: Solution at $\mu = 15$: nearly flat until the final few percent of the interval.


Visual verification   residual 3.98e-03   [VERIFIED]
both sides overlaid — they must coincide
LHS y''
RHS 15*sinh(15*y)
LHS alone: y''
RHS alone: 15*sinh(15*y)
|LHS - RHS| max = 4.05e-03
1e7
x

A stability check, for free

We also ran $\mu = 10$ a second time at a different resolution (`iterations=30`). The recovered slope:

Table 3: Recovered slope across two independent configurations

Run	$y'(0)$
defaults (`iterations=20`, 24 segments)	0.000358337784**721**
`iterations=30`, 14 segments	0.000358337784**683**

Two different resolutions, two different segmentations — agreement to roughly ten significant figures. A quantity that drifts as you refine is an artifact of the method; one that sits still is a property of the problem.

What this problem shows

- **Zero tuning is a real mode, not a demo.** At $\mu = 10$, with nothing specified, SIRPY chose its own resolution, sized its own segments, and delivered a relative residual of 2.5×10^{-10}.
- **Residuals are reported against the equation's own scale.** Absolute defect, the scale it should be judged by, and the pointwise relative residual — all three, so the reader can judge rather than trust.
- **The boundary layer is explained, not just survived.** The singularity at $x = 1.00135$, located and reported, *is* the reason Troesch's problem is hard — and the interval is confirmed clear of it.
- **When it could not solve, it said so.** At $\mu = 15$ with defaults: a stated divergence, a refusal to certify, and an explicit "NOT a solution." One parameter later, the problem is solved and verified.
- **The answer is stable.** The recovered slope agrees across independent configurations to ten figures.

A note on scope

The $\mu = 15$ default run is published here as a failure, deliberately. We consider a solver's behaviour at the edge of its defaults to be part of its specification: the requirement is not that defaults solve everything, but that the solver never certifies what it has not verified. That requirement held.

The problem library

This entry is part of the SIRPY Problem Library — an ongoing, public catalog of mathematical systems solved and verified by SIRPY.

- **GitHub** — github.com/Mitra-Institute-of-Education-MIE/sirpy-problem-library
- **Web** — mitrainstituteofeducation.org/sirpy

Submit a problem

Have a differential or integral equation you believe is challenging, unusual, or simply beautiful? We would like to see it. We will run it through SIRPY and report honestly what happened — if it solves it, how; if it struggles, why; if it reaches a current limit, we will say so. This entry is the proof that we mean it.

Contact: info@mitrainstituteofeducation.org · mitrainstituteofeducation.org/contact

Please only send problems you are free to share publicly.

SIRPY is in final validation prior to its first public release. Every line quoted above is verbatim from the solver's output; nothing is reproduced from memory.